Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.3, Jun. 2016 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2016.26.3.645

파일 I/O Interval을 이용한 랜섬웨어 공격 차단 방법론*

윤 정 무,† 조 제 경, 류 재 철‡
충남대학교

Methodology for Intercepting the Ransomware Attacks Using File I/O Intervals*

Jung-moo Youn,† Je-geong Jo, Jae-cheol Ryu‡
Chung-Nam National University

요 약

랜섬웨어는 1999년에 처음 만들어 졌지만 우리나라에서는 2015년부터 그 존재가 많이 알려지기 시작했다. 정보통신기술이 점점 발전하고 컴퓨터의 저장용량이 더욱 커지면서 컴퓨터가 저장하는 정보들이 증가했고 이 정보들을 효율적으로 관리하고 보관하는 것이 중요해졌다. 이런 상황에서 랜섬웨어는 타인의 컴퓨터에 무단으로 침입하고 정보를 담은 파일들을 컴퓨터 사용자의 허락 없이 임의로 암호화하기 때문에 사용자에게 심각한 악영향을 끼친다. 본 논문은 커널에서 특정 프로세스가 파일에 접근하는 것을 모니터링하고, 모니터링 한 정보를 바탕으로 파일에 접근하는 행위가 비정상적으로 일어났는지 탐지한다. 탐지한 결과를 통해서 특정 프로세스의 파일접근권한을 차단한다. 이러한 방법을 통해서 랜섬웨어가 비정상적으로 파일에 접근하고 암호화하는 행위를 차단하는 방법을 제시하고자 한다.

ABSTRACT

Ransomware was first created in 1999, but its existence become widely known in Korean by 2015. As information and communication technology have developed, the storage capacity of computer has enlarged, it accordingly is getting more important to effectively manage these information, rather than the information itself. In such situation, the ransomware break into other people's computer and encrypt an files without a user's permission. So, it adversely affect the user. In this paper, we monitor an access of a specific process to the file. And on the basis of this monitoring information, we detect whether the abnormal approach happened. Through the detection result, we block the permission about access to the file for a specific process. Using this method, we propose a blocking technique for the ransomeware's abnormal approach and encryption to the files.

Keywords: Ransomware, Detection, Block

I. 서 론

현대사회는 컴퓨터의 발달로 무수히 많은 정보를 사람들의 편의에 맞게 제공한다. 그러나 컴퓨터가 가지는 정보의 중요도가 높아지고 양이 많아짐에 따라서 개인정보 침해에 대한 위험성 또한 증가하고 있

Received(03. 22. 2016), Modified(06. 08. 2016), Accepted(06. 08. 2016)
* 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단 차세대정보·컴퓨팅기술개발사업(No. NRF- 2014M3C4A7030648)의 지원과 정보통신기술진흥센터의 SW컴퓨팅산업원천기술개발사업[R0190-15-2009, 화이트리스트와 상황인지 기술을 이용한 엔드포인트 보호기술 개발]의 일환으로 수행하였습니다.
† 주저자, jmstar1@cnu.ac.kr
‡ 교신저자, jcryou@home.cnu.ac.kr(Correspondingauthor)

다. 가장 대표적인 보안의 위험 요소로는 악성코드(malware)가 있으며 악성코드는 악의적인 목적을 위해 작성된 실행 가능한 코드의 통칭이다. 악성코드는 자기 복제 능력과 감염 대상 유무에 따라 바이러스, 웜, 트로이목마, 랜섬웨어 등으로 분류된다[1]. 그 중에서도 랜섬웨어란 인터넷 사용자의 컴퓨터에 잠입해 내부 문서나 스프레드시트, 그림파일 등을 암호화하여 열지 못하도록 만든 후 돈을 보내주면 해독용 열쇠 프로그램을 전송해 준다며 금품을 요구하는 악성 프로그램이다[2].

랜섬웨어는 1999년 조셉 팝(Joseph Popp)에 의해 처음 만들어 졌으며 2013년에는 비트코인이라는 디지털 통화를 사용하는 크립토락커(cryptolocker)가 출현 하였다[3]. 비트코인이란 지폐나 동전과 달리 물리적인 형태가 없는 온라인 가상화폐이다. 차세대 사이버 보안이슈와 위협 및 대처 방안에 따르면[4] 비트코인은 거래의 익명성이 보장되고 별다른 규제가 적용되지 않아서 불법적인 거래에 활용 될 수 있다고 한다. 비트코인의 익명성을 살린 크립토락커는 주로 이메일 첨부를 통하여 구 버전의 인터넷 익스플로러, 플래시의 취약점을 공격하며 컴퓨터를 감염시킨다.

운영체제 및 웹브라우저에 대한 통계를 제공하는 statcounter에 따르면 2016년 2월 컴퓨터 운영체제 점유율은 1위 Win7(46.16%) 2위 Win10(14.76%), 3위 Win8.1(11.41%), 4위 OS X(9.30%), 5위 WinXP(7.62%), 6위 Win8(3%)로 마이크로소프트사에서 만든 운영체제가 전체의 약 83%점유율을 차지하고 있으며 애플사의 운영체제는 전체의 약 9%점유율을 차지하고 있다[5]. 얼마 전까지만 해도 랜섬웨어는 마이크로소프트사의 운영체제를 사용하는 x86컴퓨터를 대상으로 감염시켰으나 2015년 자바스크립트 기술의 일종인 NW.js로 만들어진 Ransom32라는 랜섬웨어의 등장으로 리눅스나 OS X운영체제를 사용하는 컴퓨터도 더 이상 안전하지 않게 되었다[6].

Ransom32의 감염경로는 스팸메일을 이용하여 자동으로 압축이 풀리는 RAR파일을 보내는 것이다, 스팸메일을 사용자가 열어보는 순간 자동으로 랜섬웨어가 실행되며 “jpe”, “mp3”, “mov”, “docx”, “csv”, “xml”, “dat”, “pptx”등의 확장자를 가진 파일들을 128비트 AES방식으로 암호화하고 비트코인을 통해 몸값을 요구한다. 피해자가 일정 기간 동안 몸값을 지불하지 않는 경우 몸값을 더 올리거나 파일을 삭제한다.

랜섬웨어에 감염된 경우 피해자는 2가지 선택권이 있다. 제시된 몸값을 지불하거나 파일 접근 권한을 상실 하는 것이다. 암호화된 파일을 복호화 키 없이 해독하는 것이 거의 불가능하므로 암호화된 파일이 꼭 필요한 피해자는 몸값을 지불 할 수밖에 없다. 글로벌 사이버위협연합(CTA)보고서에 따르면 랜섬웨어인 크립토월로 인한 피해액은 전 세계적으로 약 3700억원에 달하는 것으로 추산된다고 한다[7]. 또한 S사가 2015년에 발생한 모바일 랜섬웨어를 분석한 결과 2014년 대비 6배 이상 증가한 약 956억원으로 집계됐다고 밝혔다.

본 논문에서는 랜섬웨어로 인한 피해를 줄이고자 랜섬웨어의 대응 방법론을 제시하고자 한다. 운영체제의 파일 I/O를 제어하는 기술을 이용할 경우 랜섬웨어를 포함한 프로세스의 파일 처리 요청을 모니터링할 수 있으며 이를 통하여 랜섬웨어 여부를 판단, 제어하는 기술을 제시하고자 한다.

II. 관련 연구

2.1 랜섬웨어

2.1.1 랜섬웨어의 개념 및 역사

랜섬웨어란 몸값을 뜻하는 ransom과 제품을 뜻하는 ware의 합성어로 사용자의 동의 없이 컴퓨터에 불법으로 설치되어 문서나 스프레드시트, 그림 파일등을 암호화하여 열지 못하도록 한 뒤, 암호화된 파일을 원상복구 시켜주는 조건으로 금품을 요구한다. 대표적인 랜섬웨어는 시놀락커(synolocker), 나부터(nsblocker), 크립토락커(cryptolocker), 크립토월(crpytowall), 테슬라크립토(teslacrpyto), 랜섬32(ransom32)등이 있다[8].

처음 존재가 알려진 랜섬웨어는 1989년 조셉 팝에 의해 만들어 진 AIDS 트로이목마다. 소프트웨어의 일부분이 라이센스가 만료되었다고 거짓 알림창을 띄우며 파일들을 암호화 한다. 파일을 해독하려면 ‘PC cyborg corporation’에 189달러를 지불 할 것을 요구한다. 2006년 중반에 발견된 gpcode, archiveus등의 랜섬웨어는 이전의 랜섬웨어보다 복호화가 더 힘들도록 RSA 암호화 알고리즘을 사용하고 확장된 키 길이를 사용했다. GpcodeAG는 660비트 RSA공개키를 이용해서 암호화했고 변종인

Gpcode.AK는 1024비트 RSA키를 이용해 암호화 하였다. 2014년에는 NAS를 대상으로 하는 랜섬웨어인 synolocker가 확산되었다.

2013년 7월에는 OS X운영체제에 특화된 랜섬웨어가 발생했다. 이것은 사용자가 불법포르노동영상을 다운로드했기 때문에 사용자를 고발한다는 웹페이지 형식의 알림창을 띄웠다. 윈도우 기반의 유사 제품과 다르게 전체 컴퓨터 성능을 저해하지 않았고 웹 브라우저의 동작을 이용했다.

2.1.2 랜섬웨어의 감염경로 및 파일 암호화 방법

한국에서는 2015년부터 급격히 랜섬웨어가 발견되기 시작했으며 보안이 취약한 사이트 및 이메일을 통하여 감염되었다. 이메일, 인스턴트 메세지, 웹사이트 등에서 링크를 클릭하면 자동으로 설치를 하고 내부에 잠입하는 방식을 사용하였다.

cryptolocker가 사용하는 알고리즘은 복합 암호화방식(hybrid encryption)이다. 파일암호화는 대칭키 암호화 방식인 AES(Advanced Encryption Standard)를 이용하고 암호화한 내용들을 복호화 하는데 쓰이는 개인키를 다시 RSA비대칭키 암호화 방식으로 암호화 한다. 이런 방식을 채택한 이유는 폴더를 탐색한 후 파일을 암호화하는 과정에서 비대칭키 암호화알고리즘(RSA)를 사용하기에는 속도가 너무 느리기 때문이다. 즉, AES암호화방식은 암호화 하는 파일의 수만큼 수행되고 RSA암호화는 한번만 수행된다. 이렇게 RSA로 암호화된 AES키와 피해자를 식별 할 수 있는 컴퓨터 정보 등과 함께 데이터베이스화 하여 관리한다. 일반적으로 256비트 AES, 2048비트 RSA암호화 알고리즘을 사용한다.

AES알고리즘으로 파일을 암호화 하면 복호화 키가 생성되는데 이를 이용하면 암호화된 파일을 복호화 할 수 있다. 컴퓨터가 cryptolocker에 감염 된 후에 악성코드를 수집하고 역 공학 분석을 통해서 암호화 하는 과정에서의 개인키 값을 추출 할 수는 있다. 하지만 대칭키 자동 생성 알고리즘의 특성 상 암호화 할 당시와 동일한 키가 생성되지 않는 것이 문제다. 그러므로 파일을 복호화하기 위해서는 처음에 암호화 후 생성된 복호화 키가 반드시 필요하다. RSA암호화 방식 또한 역 공학 분석 또는 공격자서버로 전송되는 키를 수집하더라도, 공격자가 가지고 있는 개인키를 알아내지 못하면 암호화된 파일의 복호화는 불가능하다.

2.1.3 백신의 랜섬웨어 탐지

컴퓨터가 랜섬웨어에 감염 된 경우, 한동안 CPU 팬이 매우 빠르게 회전하고 메모리를 과하게 쓰거나 때때로 화면 왼쪽 상단에 이상한 글자가 한 글자씩 생겨난다. 랜섬웨어는 컴퓨터 파일들을 탐색하기 시작하며 성능에 따라 약 5분에서 길게는 1시간정도 암호화작업을 한다. 이때까지는 컴퓨터가 조금 느려지는 것 외에는 체감하지 못한다. 파일들의 암호화가 시작되면 시스템 자원을 최대한 끌어서 최대한 빠르게 많은 파일들을 암호화하기 때문에 컴퓨터속도가 매우 느려지는 것을 체감 할 수 있다. 파일의 암호화가 완료된 후 컴퓨터를 재부팅 하게 되면 악성코드가 랜섬웨어에 걸렸음을 나타내는 알림창을 띄운다. 그 후로 감염된 컴퓨터로 일반적인 작업은 불가능하다.

2.2 특정 확장자 복구를 통한 랜섬웨어 피해 최소화 방법

C업체에서 제공하는 A제품은 랜섬웨어의 파일 암호화로 인한 피해를 최소화 할 수 있다. 대피소[9]라는 기능을 통하여 파일이 암호화 된 경우 자동으로 백업 폴더에서 파일들을 복구하는 기능이 있다. 대피소는 지정된 확장자명을 가진 파일 전부를 A제품에서 보호하는 특정 폴더에 백업을 시켜놓는 것이다. 랜섬웨어가 파일을 암호화 할 때 파일의 offset에 접근하는데, A제품에서 파일의 offset에 접근하는 프로세스를 감지하면, 해당 파일을 백업한다. A제품이 랜섬웨어의 파일 암호화를 탐지하면 “랜섬웨어 행위 탐지” 팝업창을 띄우고 악성 프로세스를 차단함과 동시에 암호화 된 파일들을 대피소에 저장된 원래 파일로 복구한다. 랜섬웨어의 파일 암호화동작에 대한 탐지는 국내 S사에서 운영하는 멀웨어즈닷컴(malwares.com)API를 추가하여 실행파일에 대한 검사를 수행한다.

2.3 미끼 파일을 이용한 랜섬웨어 사전 탐지 방법

E사의 A제품은 컴퓨터 하드디스크에서 각 드라이브별로 숨김 속성 값을 가지는 폴더를 생성하며, 폴더 내부에는 4종의 미끼파일인 document.doc, hancom.hwp, image.jpg, text.txt파일이 추가된다. 즉, A사의 랜섬웨어 차단 방식은 랜섬웨어가 미끼 파일을 암호화하면, 이를 탐지하여 알림창을 사용자에게 보여준다.

앞서 설명한 랜섬웨어의 대비책은 랜섬웨어를 차단하기 보다는 랜섬웨어가 암호화한 파일을 원래의 파일로 복구를 하거나, 가짜 파일을 생성하여 랜섬웨어가 바꾸는 시점을 탐지하도록 되어 있다. 이는 랜섬웨어의 근본적인 해결책이기보다는 피해 최소화 방안이라고 할 수 있다. 특히 가짜 파일을 이용한 방법은 악성코드가 해당 파일을 암호화 하지 않는 방법으로 얼마든지 우회 할 수 있다. 따라서 본 연구에서는 근본적으로 차단이 가능한 방법으로 커널 기반의 파일 I/O 모니터링 및 제어 기술을 이용하고자 하며, 사용자가 수행할 가능성이 매우 낮은 파일 I/O 행위에 대하여 랜섬웨어로 판단하고 차단하고자 한다.

III. 제안 연구 방법

랜섬웨어의 특정 프로세스가 파일 암호화를 시도할 때, 프로세스는 운영체제의 커널에 파일 I/O를 요청하게 된다. 본 연구에서는 이러한 파일 I/O를 탐지하고, 탐지한 파일 I/O 정보를 바탕으로 이상징후 탐지모델이 파일의 접근이 비정상적인지 판단한다. 파일에 대한 접근이 비정상적일 경우, 랜섬웨어로 판단하고 해당 프로세스의 파일접근권한을 차단하여 파일 암호화를 불가능하게 한다.

3.1 커널을 이용한 파일 I/O 모니터링 및 제어 모델

커널을 이용한 파일 I/O 모니터링 및 제어 모델은 다음과 같다.

프로세스의 행동을 탐지하고 제어하기 위해서 커널의 기능을 활용한다. 제안하는 본 연구의 방법론은 커널영역에서 동작하므로 유저레벨에서 동작하는 프로그램들에 비해 권한이 높고, 더 효율적으로 동작하게 된다.

랜섬웨어는 파일을 암호화하기 위해서 프로세스를 생성하여 파일을 탐색하고, 파일을 변경(암호화)하게 된다. 심지어 파일의 확장자명까지 변경하는 랜섬웨어도 있는 것으로 확인되었다. 이러한 파일의 변경 행위는 커널에서 탐지 및 제어가 가능하며 이는 마이크로소프트사의 윈도우즈 제품군에서는 드라이버라는 개념으로 제공하며, 애플사의 OSX 제품군에서는 커널 익스텐션(kernel extension)이라는 개념으로 제공하고 있다. 본 논문에서는 이러한 개념을 커널 모듈이라는 통칭 하에 사용하고자 한다.

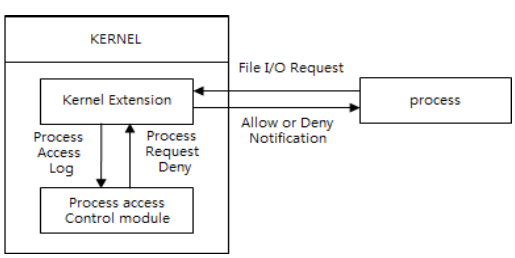

Fig. 1. File I/O Monitor and Control model

3.1.1 파일 I/O 모니터링 및 제어 모델

파일 I/O 모니터링 모델은 커널영역에서 특정한 확장자명을 가진 파일에 대한 처리 요청을 탐지한다. 파일 처리를 요청한 프로세스가 파일내용을 변경하기 위해서는 반드시 파일을 열어야하며, 커널은 파일을 열었을 때 발생하는 신호를 커널 모듈에 전달함으로써 해당 프로세스가 파일에 접근한 시간 및 파일경로를 커널 모듈에서 확인 할 수 있다. 또한 파일 I/O 모니터링 모델은 프로세스의 파일 접근에 대한 탐지뿐만 아니라 프로세스의 파일 접근 제어가 가능하다. 특정한 프로세스 인식자(process identifier, 이하 pid)값을 통하여 각각의 프로세스를 식별 가능하며, 이를 이용하여 특정 프로세스의 파일접근을 제어 할 수 있다. 그리고 pid와 상관없이 특정한 파일을 기준으로 파일에 접근하려는 프로세스를 제어 할 수 있다.

파일 I/O 모니터링 모델은 파일에 접근한 프로세스의 정보를 수집하고 제어하는 역할만 수행하며, 파일접근 자체가 합법적인지 불법적인지에 대한 판단은 하지 않는다. 파일 I/O 모니터링 모델은 정보 송수신 모델로부터 받은 정보를 이용해 랜섬웨어의 프로세스로 인지하고 해당 프로세스의 파일접근권한을 차단한다.

3.1.2 파일 접근정보 송수신 모델

커널에서 동작하는 파일 접근정보 송수신 모델은 차단해야 할 프로세스에 대한 정보와 파일 처리 요청에 대한 정보를 파일 I/O 모니터링 및 제어모델과 공유한다. 이 정보는 소켓을 이용해 유저영역에서 동작하는 Detector와 주고받는다. 소켓으로 송수신하는 메세지의 내용은 파일 I/O 모니터링 및 제어모델이 특정한 프로세스가 파일에 접근할 경우 탐지했던 pid와 시간, 파일경로가 있다. 소켓은 해당 정보를 보관하고 있다가 Detector와의 연결이 확인되면

Detector에게 메세지를 송신한다.

반대로, 파일 접근정보 송수신 모델은 Detector로부터 메세지를 수신 할 수도 있다. 최초 정보 송수신 모델이 Detector에 데이터 메세지를 송신한다. Detector는 수신한 메세지를 열어서 특정 파일에 접근하려는 프로세스를 확인하고 접근이 정상적인지 비정상적인지를 판단한다. 프로세스가 비정상적으로 파일에 접근한다고 판단하면, Detector는 파일 접근 정보를 정보 송수신 모델에게 송신한다. 정보 송수신 모델은 수신한 메세지를 파일 I/O 모니터링 및 제어 모델과 공유한다. 결과적으로 파일 I/O 모니터링 및 제어 모델이 랜섬웨어가 사용하는 프로세스에 대한 파일접근을 차단함으로써 파일 암호화를 불가능하게 만든다.

3.2 비정상적 파일접근 탐지 모델(Detector)

비정상적 파일 접근 탐지 모델은 다음과 같다.

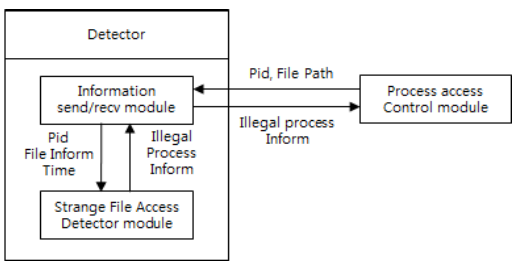

Fig. 2. Strange File I/O Detector model

3.2.1 파일 접근 정보 수신 및 파일 접근 명령 송신 모델

특정 프로세스가 커널에게 파일 접근을 요청 할 경우 파일 I/O 모니터링 및 제어모델은 관련 정보를 담은 메세지를 Detector에게 송신한다. 구체적으로, 커널에서 작동하는 파일정보 송수신 모델은 소켓을 가지고 있다. 이 소켓을 이용하여 사용자 레벨에서 작동하는 Detector의 파일정보 송수신 모델이 가지고 있는 소켓에게 정보 메세지를 송신한다. 이 모델은 지속적으로 파일 접근정보 메세지를 수신하고 메세지 내용을 이상징후 탐지모델과 공유한다. 리시브(recv)함수를 통해 수신한 이 정보 메세지는 정보를 문자열타입의 변수를 이용하여 보관한다. 메세지에 담을 정보의 최대 길이는 1024바이트로 설정했다. 정보 메세지는 pid, 파일접근시간, 파일경로가 들어있다. pid, 파일접근시간 및 파일경로는 가변적인 길이를 가진다. 하지만 pid 와 파일접근시간은 포맷이 정해져 있기 때문에 어느 정도 길이가 정해져있다. 운영체제마다 파일 최대경로길이가 다르며 윈도우 운영체제 같은 경우에는 폴더 및 파일의 전체 경로길이를 담는 변수의 최대 길이는 260바이트이다. 따라서 일반적인 사용자는 하위폴더를 무수히 많이 만들어 사용한다 하더라도 파일경로의 길이까지 합친 사이즈가 1024바이트를 넘는 경우가 거의 없다. 커널과 주고받는 데이터양이 많아질수록 커널에서 과부하로 인한 시스템 에러를 유발 할 가능성이 커기 때문이지기 때문에 메시지의 크기를 1024바이트로 제한하였다.

Detector의 정보 송수신 모듈이 수신한 메세지는 텍스트파일에 수신한 내용을 저장한다. 텍스트파일에 저장하지 않고 실시간으로 메세지를 확인해서 비정상적인 파일접근을 판단 할 수도 있지만 로그를 남겨서 파일 접근판단에 대한 근거자료를 가지고 있는 것이 추후 에러가 발생한 경우에 더 능동적으로 대처 할 수 있다.

커널의 정보송수신 모델이 메세지를 송신하는 시점을 모르기 때문에 Detector의 정보 송수신 모델은 계속 대기하면서 메세지를 수신 할 때마다 텍스트파일에 메세지를 추가한다.

3.2.2 이상징후 탐지 모델

이상징후 탐지 모델은 앞서 설명한 파일 접근 정보 수신 및 파일 접근 명령 송신 모델에서 저장한 정보를 이용하여 판단한다. PID, 파일 접근 시간, 파일의 전체 경로에 대한 정보를 구조체 형태로 저장하고 있으며, 이를 이용할 경우 빠른 검색 및 비교가 가능하다.

하지만 파일에 대한 I/O 요청은 랜섬웨어의 프로세스뿐만 아니라 일반적인 프로세스에서도 요청하기 때문에 단순히 한 번의 파일 I/O 요청만으로는 비정상적인 접근을 판단 할 수 없다. 이상징후 탐지 모델은 가장 최근에 저장된 10개의 메세지를 확인한다. 10개의 메세지 중에서 동일한 프로세스가 5개 이상이면 동일한 프로세스의 메세지들을 가져온다. 가져온 메세지 중에서 가장 최근의 메세지와 가장 오래된 메세지의 시간차이가 0.01초 이하일 경우에 비정상적인 접근으로 판단한다. 비정상적인 파일접근으로 판단 할 경우, 정보 송수신 모델과 정보를 공유하고

이 정보를 담은 메세지를 커널의 정보 송수신 모델에 전송한다. 메세지를 수신하면 해당 정보를 파일 I/O 모니터링 및 제어 모델이 확인하고 해당 프로세스의 파일 접근권한을 차단한다.

이상징후 탐지 모델은 pid를 기준으로 비정상적으로 파일을 접근한 프로세스의 명령어 또는 경로를 확인하고 pid와 관련 정보를 사용자에게 알려준다. 만약 비정상적인 파일접근을 하는 프로세스가 실제로 랜섬웨어의 프로세스라면 사용자에게 차단 여부를 묻는 동안에도 파일암호화가 진행되기 때문에 프로세스를 우선 차단 한 후에 사용자에게 차단여부를 묻는다. 이상징후 모델은 비정상적으로 파일에 접근하는 프로세스를 사용자에게 알리고 차단여부를 묻는 역할을 한다. 사용자가 차단을 원하지 않은 경우 이상징후 탐지모델은 정보 송수신 모델에게 차단초기화 메세지송신을 요청한다. 이 메세지를 커널영역의 정보 송수신 모델이 수신하면 파일 I/O 모니터링 탐지 및 제어 모델과 메세지내용을 공유하며, 파일 I/O 모니터링 탐지 및 제어 모델은 차단하고 있던 프로세스의 파일접근 권한을 원래 상태로 되돌려 놓는다.

IV. 실 험

본 연구에서 제안한 랜섬웨어 차단 방법론은 악성코드의 파일 암호화를 커널에서 제어하는데 기반하고 있다. 현재 커널을 통한 파일 I/O 처리는 마이크로소프트사의 윈도우즈 제품군에서 활발하게 이루어지고 있지만 최근 점유율이 계속 상승하고 있는 Apple 사의 OSX 환경을 이용하여 수행하였다. 특히 윈도우즈 제품군의 제어 방법은 많이 공개되어 있지만 OSX 환경에서의 제어 방법은 많이 공개되어 있지 않아, 본 연구를 통하여 OSX 환경에서의 안전성을 높이는데 많은 도움이 되고자 한다.

가상의 OSX 운영체제를 구축 한 후 랜섬웨어인 cryptolocker의 샘플 두 개를 확보해서 실제로 랜섬웨어를 작동시켰다. 랜섬웨어의 프로세스가 실행되면 지정된 확장자를 가진 파일을 검색하고, 검색한 파일들을 암호화한다. 이 때, 해당 프로세스가 파일에 접근한 이벤트들을 200개 단위로 묶어서 시간을 측정했다. 파일접근 이벤트를 통해 실험에 사용된 cryptolocker 샘플은 “txt”, “pdf”, “pptx”, “dat”의 확장자를 가진 파일을 암호화했다. 실험환경으로는 운영체제는 OS X 10.10.5 Yosemite을 사용하고 2.67Ghz의 CPU, 2GB의 메모리를 사용했다. “pptx“, ”txt“의 확장자를 가진 파일들을 암호화에 사용했다. 구체적으로, 76KB에서 최대 69.2MB의 크기의 ”pptx“파일을 사용했고, ”txt“파일의 경우 152KB이하의 파일을 사용했다. 실험 시작 전 CPU 사용량은 5%이하로, 실험에 불필요한 프로그램들은 종료하였다. 먼저, 첫 번째 cryptolocker샘플을 사용한 실험 결과는 Table 1과 같다.

가상환경의 운영체제에서 파일암호화진행시 랜섬웨어의 프로세스는 총 3600번 파일에 접근했다. 한 번의 접근을 한 번의 파일 암호화라고 본다면, 200개의 파일을 암호화 하는데 소모되는 시간은 작게는 0.07초에서 길게는 20.94초 소모된다. 제안한 연구방법은 가장 최근에 발생한 10개의 파일접근 이벤트에서 한 프로세스가 5개 이상의 파일에 접근 한 시간차이를 이용하므로 매 실험 단위마다 5개의 파일을 암호화하는데 걸리는 평균 시간을 측정했다. 측정결과, 가장 작게는 0.00175초가 걸렸고 가장 길게는 0.7075초가 걸렸다. 위의 실험을 기준으로 5개의 파일 접근에 대하여 적어도0.00175초 차이가 날

Table 1. The first cryptolocker sample's File Encryption

experiment	lead time	5 file encryption average
1	8.15	0.20374
2	5.69	0.14225
3	0.75	0.01875
4	0.07	0.00175
5	0.07	0.00175
6	0.66	0.0165
7	20.94	0.5235
8	18.27	0.45675
9	0.26	0.0065
10	0.6	0.015
11	2.83	0.7075
12	0.81	0.02025
13	1.08	0.027
14	2.34	0.0585
15	3.51	0.08775
16	2.64	0.066
17	2.32	0.058
18	2.75	0.06875

경우를 탐지해야만 암호화 할 대상 파일 모두에 대한 탐지가 가능하다.

파일접근 이벤트들을 일일이 확인한 결과 "txt"의 확장자를 가진 텍스트파일들을 암호화하는 데 걸린 시간이 나머지 확장자를 가진 파일들을 암호화 하는 데 걸리는 시간에 비해서 월등히 빠르게 암호화 되는 것을 알 수 있었다. 그래서 텍스트파일을 빼고 다시 동일한 실험을 진행하였다. 결과는 Table 2와 같다.

텍스트 파일을 제외한 100개의 파일을 대상으로 실험을 진행했고 10개의 파일을 하나의 실험단위로 묶어서 시간을 측정했다. 5개의 파일을 암호화 하는 데 걸리는 평균 시간은 0.04초에서 최대 10.87초가 소모됐다. 텍스트 파일을 제외하고 나머지 파일들에 대해서만 탐지 할 경우에는 기존의 0.00175의 시간 차이를 탐지하는 것에서 0.04초의 시간을 탐지하는 것으로 탐지 요구 시간이 늘어나는 것을 확인했다.

위와 동일한 실험을 두 번째 crpytolocker 샘플을 사용하여 진행했다. 텍스트 파일을 포함하여 파일 암호화를 진행했을 때 5개의 파일을 암호화하는데 걸리는 평균시간이 최소 0.00175초에서 최대 0.4685초 소모되는 것을 확인했다. 텍스트파일을 제외하고 파일암호화시간을 측정했을 때, 최소 0.02초에서 최대 6.715초가 소모되었다. 실험결과는 다음과 같다.

두 번째 cryptolocker샘플을 사용하면 텍스트파일을 포함했을 경우 최소 0.00175초의 탐지시간이 필요하고 텍스트파일을 제외하면 0.02초의 탐지시간이 필요하다. 위의 두 개의 랜섬웨어 샘플을 사용하여 실험을 했을 때, 텍스트파일을 제외하면 최소한 0.02초 동안 5개의 파일에 대하여 접근을 시도 한 경우를 반드시 탐지해야만 랜섬웨어의 프로세스로 인식 할 수 있다. 하지만, 위의 실험에서는 5개의 파일

Table 2. The first cryptolocker sample's File Encryption excepting .txt Files

experiment	lead time	5 file encryption average
1	1.97	0.985
2	3.01	1.505
3	0.08	0.04
4	0.4	0.2
5	0.07	0.035
6	0.3	0.15
7	0.03	0.015
8	20.45	10.225
9	21.74	10.87
10	0.83	0.415

Table 3. The second cryptolocker sample 's File Encryption

experiment	lead time	5 file encryption average
1	8.26	0.2065
2	6.13	0.15325
3	0.83	0.02075
4	0.07	0.00175
5	0.08	0.002
6	0.54	0.0135
7	0.73	0.01875
8	18.74	0.4685
9	0.26	0.0065
10	0.58	0.0145
11	2.76	0.069
12	0.84	0.021
13	1.06	0.0265
14	8.95	0.22375
15	2.62	0.0655
16	4.83	0.12075
17	2.26	0.0565
18	2.83	0.07075

Table 4. The second cryptolocker sample's File Encryption excepting .txt Files

experiment	lead time	5 file encryption average
1	2.49	1.245
2	13.43	6.715
3	0.07	0.035
4	0.23	0.115
5	0.09	0.045
6	0.2	0.1
7	0.04	0.02
8	0.17	0.085
9	11.99	5.995
10	0.6	0.3

을 암호화 하는데 걸리는 평균시간이 기준이므로 0.02초의 시간보다 더 작게 시간차이가 나서 랜섬웨어 프로세스를 탐지하지 못 할 경우도 있을 것으로 판단되며 지속적인 실험을 통해 최적화 된 기준을 찾을 필요가 있다. 본 연구에서는 텍스트파일을 포함하여 파일 입출력을 탐지하면 커널모듈에 많은 과부하를 주기 때문에 텍스트 파일을 제외하고, 한 프로세스가 5개의 파일에 대하여 접근하는 시간이 0.02초의 절반인 0.01초 이하일 경우에 대하여 랜섬웨어의 프로세스로 판단하기로 했다.

애플사의 OS X 운영체제에서 랜섬웨어샘플을 작동시켰다. 파일 I/O모니터링 및 제어모델로부터 메세지를 수신한 Detector는 파일접근정보를 메세지 단위로 저장했다. 가장 최근에 저장한 10개의 메시지 정보를 가져와서 그 중 동일한 프로세스의 파일접근이 5개 이상인지 확인한다. 5개 이상일 경우, 가장 오래된 메세지와 가장 최근의 메세지의 시간차이가 0.01초 이하면 랜섬웨어의 프로세스로 판단하고 해당 프로세스정보를 담은 메세지를 파일 I/O모니터링 및 제어모델로 송신했다. I/O모니터링 및 제어모델은 수신한 메세지의 정보를 확인하고 프로세스의 접근권한을 성공적으로 차단했다. 0.01초라는 시간차이는 커널과부하를 고려한 수치이므로 텍스트파일은 실험에서 배제했다. 그 외의 파일에서는 효과적으로 탐지 및 제어가 가능했다. 물론, 5개의 파일에 대한 시간차이로 랜섬웨어의 프로세스를 탐지하기 때문에 랜섬웨어의 프로세스가 처음 암호화 한 5개의 파일에 대해서는 파일암호화를 막지 못했다. 하지만 5개의 파일을 제외한 나머지파일은 랜섬웨어프로세스의 파일접근권한을 차단함으로써 랜섬웨어로부터 안전 할 수 있었다. 랜섬웨어가 암호화하는 파일 중에서는 해당 파일을 사용하기 위해서 특정 프로그램을 필요로 하기도 한다. 이 프로그램들을 설치만 해도 적어도 5개 이상의 특정 확장자의 파일이 기본적으로 설치되기 때문에 최초 암호화된 5개의 파일이 사용자가 생성한 파일이 아닐 수도 있다. 실험결과에 의하면 텍스트파일이나 처음 탐지한 5개 파일에 대해서는 암호화를 막지 못했으나 그 외의 경우에는 암호화를 막을 수 있었다. 랜섬웨어로부터 피해를 최소화하기 위해 희생된 파일들은 백업을 이용한 암호화 무력화 연구가 추가적으로 필요하다.

V. 결 론

실험에서 볼 수 있듯이, 커널영역에서 랜섬웨어 프로세스가 파일에 접근하는 것을 확인하고 해당 프로세스의 추가적인 파일접근정보를 통해서 성공적으로 랜섬웨어 프로세스의 파일접근 권한을 차단 할 수 있었다. 하지만 본 연구에서의 랜섬웨어 프로세스 차단방법은 시간 간격을 기준으로 하고 있기 때문에 랜섬웨어의 암호화 기술이 점점 증가 할수록 시간간격이 더 짧아 질 수 있다는 단점이 있다.

랜섬웨어로부터 발생하는 피해가 점점 증가하고있는 상황에서 위의 연구는 반드시 필요하다. 파일암호화를 위해서 프로세스는 파일에 접근할 수밖에 없기 때문에 이것을 이용한 본 연구는 랜섬웨어의 피해를 최소화 시킬 수 있다. 위의 방법을 사용하면 100% 확률로 랜섬웨어의 파일암호화를 차단 할 수는 없지만 커널에 대한 연구와 실험을 지속적으로 해서 더 효율적으로 랜섬웨어를 탐지하고 차단해야한다. 이를 위해서는 사용자의 행위에 대한 다양한 연구가 추가적으로 필요하며 다양한 환경에서의 파일접근에 대한 연구가 지속적으로 필요하다.

References

[1] "Malware," https://en.wikipedia.org/wiki/Malware

[2] Bong-joon Kim, Woon-soo Kim, Jung-hwan Lee, Sin-hyuk Yim, Sang-geun Song, and Sang-jun Lee, "Design and Implementation of a Ransomware Prevention System using Process Monitoring on Android Platform", *The Korean institute of information scientists and engineers*, 2015(12), pp. 852-853, Dec. 2015

[3] Gates, Megan, "CYBERSECURITY As ransomware continues to spread, companies must decide whether to back up or pay up to get their data back," *The american society for industrial security*, vol. 59, no. 12, pp. 26-26, Dec. 2015

[4] Gyeong-sin Kim and Moon-sik Kang, "The next generation of cyber security issues and threats and countermeasures," *The*

institute of electronics engineers of korea, 41(4), pp. 69-77, Apr. 2014
〔5〕 "Operating System rate," http://www.koreahtml5.kr/jsp/infoSquare/browserUseStats.jsp
〔6〕 "Ransom32," http://securityaffairs.co/wordpress/43250/cyber-crime/ransom32-crypto-ransomware.html
〔7〕 Cyber THREAT ALLIANCE, "Lucrative Ransomware Attacks:Analysis of the CryptoWall Version 3 Threat," http://cyberthreatalliance.org/crypto-wall-executive-summary.pdf
〔8〕 "Ransomware Notable examples," https://en.wikipedia.org/wiki/Ransomware
〔9〕 "Shelter," http://hummingbird.tistory.com/6196

〈저자소개〉

윤 정 무 (Jung-moo Youn) 학생회원
2013년 2월: 충남대학교 컴퓨터공학과 졸업
2015년 8월~현재: 충남대학교 컴퓨터공학과 석사과정
〈관심분야〉 정보보호, 시스템보안

조 제 경 (Je-geong Jo) 학생회원
2006년 2월: 한신대학교 정보시스템공학 졸업
2008년 8월: 한신대학교 컴퓨터정보학 석사
2014년 3월~현재: 충남대학교 컴퓨터공학 박사과정
〈관심분야〉 정보보호, 시스템보안, 네트워크보안

류 재 철 (Jae-cheol Ryou) 종신회원
1985년 2월: 한양대학교 산업공학과 졸업
1988년 5월: Iowa State University 전산학 석사
1990년 12월: Northwestern University 전산학 박사
1991년 2월~현재: 충남대학교 컴퓨터공학과 교수
〈관심분야〉 정보보호, 네트워크보안, 암호학, 보안프로토콜